## CLAIMS

WHAT IS CLAIMED IS:

1      1. A method of processing network data in a network processor comprising:

2           scheduling a first thread to process a first incoming block of data; and

3           scheduling a second thread to process a second incoming block of data prior

4              to the first thread completing.

1      2. The method of claim 1, wherein the first incoming block of data and the second

2 incoming block of data are from a common data packet.

1      3. The method of claim 2 further comprising:      *interrupt.*

2           saving state information by the first thread; and

3           retrieving the state information by the second thread.

*address*
*/ center,*

1      4. The method of claim 3, wherein the state information includes a pointer into a

2 memory indicating where to move the first and second incoming blocks of data.

1      5. The method of claim 4 further comprising:

2           storing data to memory in a sequential ordering based on the state

3              information.

1      6. The method of claim 5 further comprising:

2     providing the state information to transmit circuitry.

1   7. A method of processing a data packet received over a network comprising:

2     processing a first portion of the data packet using a first thread; and

3     simultaneously processing a second portion of the data packet using a second

4      thread.

1   8. The method of claim 7 wherein the first thread and the second thread do not

2 time share processing with one another.

1   9. The method of claim 8 wherein the first thread and the second thread operate

2 out of different microengines.

1   10. The method of claim 7 wherein the first thread and the second thread time

2 share processing with one another.

1   11. The method of claim 10 wherein the first thread and the second thread operate

2 out of a common microengine.

1   12. The method of claim 7 further comprising:

2     simultaneously with processing the first portion and the second portion of

3      the data packet, processing a third portion of the data packet using a

4      third thread.

1    13. The method of claim 12 wherein the first thread, the second thread, and the

2    third thread run the same code.

1    14. The method of claim 13 wherein the first thread, the second thread, and the

2    third thread do not time share processing with one another.

1    15. An article comprising a computer-readable medium which store computer-

2    executable instructions for receiving data from a plurality of ports, the instructions

3    causing a computer to:

4        process a first portion of a data packet using a first thread; and

5        process a second portion of the data packet using a second thread, wherein

6            there is no time sharing between the first thread and the second thread.

1    16. The article of claim 15, the article further comprises instructions to:

2        save state information of the first thread; and

3        restore the state information by the second thread.

1    17. The article of claim 16, the article further comprises instructions to:

2        provide the state information to transmit circuitry when an end of packet is

3            detected by a subsequent thread.